Aligning Monitoring and Compliance Requirements in Evolving Business Networks

Marco Comuzzi

School of Mathematics, Computer Science and Engineering
City University London
Northampton Square, EC1V 0HB London, United Kingdom
marco.comuzzi.1@city.ac.uk

Abstract. Dynamic business networks (BNs) are intrinsically characterised by change. Compliance requirements management, in this context, may become particularly challenging. Partners in the network may join and leave the collaboration dynamically and tasks over which compliance requirements are specified may be consequently delegated to new partners or backsourced by network participants. This paper considers the issue of aligning the compliance requirements in a BN with the monitoring requirements they induce on the BN participants when change (or evolution) occurs. We first provide a conceptual model of BNs and their compliance requirements, introducing the concept of monitoring capabilities induced by compliance requirements. Then, we present a set of mechanisms to ensure consistency between the monitoring and compliance requirements when BNs evolve, e.g. tasks are delegated or backsourced in-house. Eventually, we discuss a prototype implementation of our framework, which also implements a set of metrics to check the status of a BN in respect of compliance monitorability.

1 Introduction

Business processes represent the foundation of all organizations and, as such, are subject to government and industry regulation. Organizations must collect data during business process execution to demonstrate the *compliance* with the regulations they are subject to. Examples of such regulations can be found in different industries and disciplines, such as HIPAA in health care, Basel II and SOX in financial management and accounting, or ISO 9001 for quality management.

Faster market dynamics and fiercer competition also push organizations to focus on their core business, engaging in Internet-enabled, highly dynamic collaborations with external partners, referred to as virtual enterprises or collaborative Business Networks (BNs) [5, 4]. BNs are characterized by cross-organizational business processes , i.e. processes that span the boundaries of individual organizations. When (part of) processes are delegated (outsourced) to other partners, organizations may loose visibility over such processes.

This becomes particularly relevant to the case of compliance requirements monitoring. As a result of delegation, compliance requirements will predicate on

tasks executed by several heterogeneous organizations. Collecting the appropriate information to monitor such requirements and keeping this aligned with the compliance requirements defined in the BN may become challenging, since organizations (i) may not want to disclose relevant information, (ii) may not know which pieces of information they should disclose, or (iii) may not be able to capture the required information in their own information systems. For instance, Section 404 of the SOX act states that data security breaches should not be hidden from auditors and always reported to shareholders [14]. A SOX-compliant company A delegating parts of its activities to a company B could remain compliant only if company B would be able to report data security breaches (at least on the data A shared with B to delegate its activities).

Altough there are standards, such as ISAE 3402, specifically targeting compliance in service organization outsourcing, these still focus on long-term one-to-one relationships between clients and external service organizations, defining the controls required by clients and the requirements of the service organization to satisfy such controls. Moreover, they take a static perspective on compliance, as they only imply the generation of ex-post reports of periodic audits.

The Internet-enabled BNs that we are considering in this work, however, are intrisincally dynamic, i.e. they are characterized by change, such as partner substitution or process outsourcing exploiting dynamic partner selection [5, 2]. In extreme cases, BNs can be *instant* [11], i.e. they are setup to tackle a specific business need, such as the ad-hoc organization of a large-scale event or managing the recovery from a natural disaster, and will be dismantled immediately after the business need has been served. In such a scenario, compliance management cannot rely on long term contracts and ex-post audits, but it should rather mimic the intrinsic dynamicity of the collaboration.

This paper proposes a framework for aligning business process compliance and monitoring requirements in dynamic BNs. Our focus, therefore, is on maintaining the alignment between process compliance and monitoring requirements as BNs *evolve*. In particular, our framework is based on the premise that compliance requirements induce monitoring requirements in the BNs. For instance, to verify the occurrence of a certain task in a process, the organization responsible for the execution of the process needs to provide some evidence of the task execution, such as an event log showing the execution of the task extracted from the company's internal information systems. Given a set of compliance requirements, our framework computes the set of actions required in the BN to maintain the monitorability of the requirements when evolution occurs. Evolution can be at the structural level, e.g. a process or part of it is outsourced from one actor to another actor in the BN, or at the compliance management level, e.g. a new compliance requirement is introduced in the BN.

The steps of the development of our framework are shown in Fig. 1. We first define a conceptual model for representing BNs, their evolution, and their compliance and monitoring requirements. At the conceptual level we also define the mechanisms for aligning compliance and monitoring requirements. The conceptual model is not directly implementable, since it abstracts from specific choices

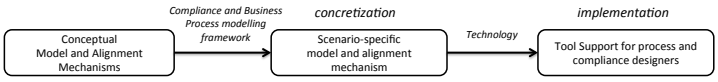

Fig. 1. Steps in framework development

made for the formalization of compliance requirements and for the modelling of business processes. Such choices are taken into account to derive the specific model and alignment mechanisms. We label this step *concretization* in our conceptual framework. Concrete models and mechanisms are implementable into a tool to support the designers in managing compliance and monitoring concerns.

The paper is organised as follows. Section 2 introduces the meta-model underpinning our framework and Section 3 presents the mechanisms to align compliance and monitoring requirements at a conceptual level. Section 4 discusses the operationalisation of our conceptual framework for specific implementation choices regarding the language to express compliance requirements and the chosen framework for business process modelling. A prototype implementing our operationalised framework is presented in Section 5, whereas related work is discussed in Section 6. Eventually, we draw our conclusions and discuss future work in Section 7.

2 Conceptual Model

The conceptual model underpinning our framework is shown in Fig. 2.

In our model, a business network is constituted by a set of business entities (actors), which participate to business processes. Participation has to be intended here at the operational level, that is, actors execute specific parts of a business process to which they participate. As such, actors may be able to provide the information required to monitor compliance of the business processes to which they participate.

Without loss of generality, we consider single-entry, single-exit, block-structured processes [19]. Hence, a process is constituted by a set of blocks. Blocks are the process decomposition unit over which we define compliance requirements. Blocks can be outsourced to another actor in the BN. In our model, the outsourcing relationship is captured at the process structural level, that is, an outsourced block points to the process to which it has been outsourced to.

We deal with processes specified using the *public view* [12, 16, 5]. The public view of a process retains those aspects that are relevant for the collaboration among partners in a BN and hides the internal detailed specification required for process enactment by individual partners (specified in the *private view*). Note that we are not concerned with the control flow of processes. Being able to monitor a set of compliance requirements, in fact, does not depend on the particular path followed by process execution. In other words, we focus on design-time compliance, i.e. designing BNs and processes that *can be* monitored [22], irrespectively of the particular path followed by process instances.

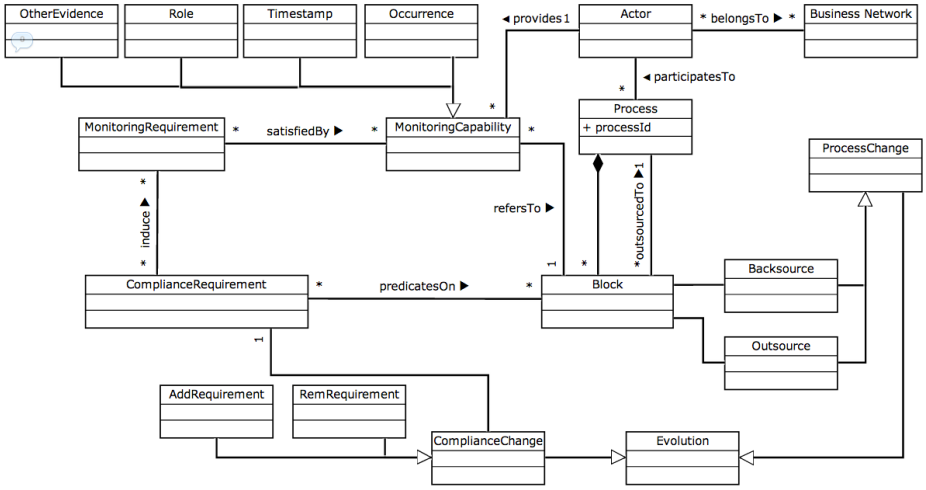

Fig. 2. Conceptual Model of BN and compliance

As far as compliance is concerned, a BN is characterized by several compliance requirements. A compliance requirement refers to any explicitly stated rule of regulation that prescribes any aspect of business processes in a BN [8, 22].

In order to be monitored, compliance requirements induce monitoring requirements, i.e., the information that actors participating to a certain process should provide to guarantee that compliance requirements can be verified. Monitoring requirements are satisfied by actors through the implementation of monitoring capabilities. A monitoring capability is conceptually defined as the ability of an actor to provide the information about a process required to check the satisfaction of one or more compliance requirement.

We identify four types of monitoring capabilities, i.e. Occurrence, Timestamp, Role, and Other Evidence. The Occurrence capability is the ability to demonstrate the execution of a certain activity or block thereof. A Timestamp capability refers to the ability of providing reliable information about the instant in time in which an activity has been executed, e.g. in the form a timestamped event. Timestamp information is required, besides evidence of the occurrence, to monitor compliance requirements referring to the ordering of activities. A Role capability refers to the ability of a business entity to provide reliable information about the role (person, department, business unit) executing a certain activity. This capability is required for instance to monitor segregation of duties. These three capabililty types capture information about the *provenance* of processes [3], i.e. what manipulation have been performed in the process, when and by whom.

Eventually, the OtherEvidence capability refers to the ability of an actor to provide evidence demonstrating that a certain activity has not occurred. While non-occurrence cannot be demonstrated unequivocally, such evidence may be

provided in the form of a list of all execution traces of a process not showing the execution of the block for which non-occurrence is required [22].

We argue that the above four types of monitoring capabilities can cover most of the monitoring requirements derived from compliance requirements expressed using LTL or other languages that can be translated into LTL, such as Event Calculus. A discussion of how the above types of monitoring capabilities cover the LTL compliance patterns considered by [8] is presented in Section 4.1.

Eventually, as far as evolution is concerned, for reasons of bervity we consider a subset of the BNs evolution types considered by [5] in the context of SLA management in evolving business networks. The evolution of a BN can occur at the level of compliance requirements or processes. New compliance requirements can be introduced (added) or existing compliance requirements may no longer be necessary for a BN (remove). Processes or, more specifically, blocks within processes, can be either outsourced by one partner to another in the BN or backsourced in-house.

3 Compliance-Monitoring Alignment Mechanisms

The mechanisms to align monitoring and compliance requirements can be conceptually defined on the basis of the conceptual model introduced in the previous section. A formal characterization of such mechanisms can only be given once specific implementation choices have been made, such as choosing the language to express compliance requirements or the way outsourcing is captured in business process models. Examples of formal characterizations of our alignment mechanisms for specific implementation choices are presented in Section 4.2.

The mechanism to align compliance and monitoring requirements when a new compliance requirement new_cr is introduced is presented in Alg. 1. In a nutshell, for all the blocks on which new_cr predicates, the *single alignment check* procedure is called. In such procedure, first we assess whether the block under consideration is outsourced. If that is not the case, then we derive the monitoring requirements induced by new_cr and, consequently, the monitoring capabilities required to satisfy such monitoring requirements. Note, in fact, that monitoring capabilities can be implemented only by the actors actually participating to a process on which the new compliance requirement predicates. If the block is outsourced, then the procedure is recursively called on the process to which the block is outsourced. Such recursive call allows the mechanism to execute over arbitrarily long outsourcing *chains*. This is consistent with techniques commonly adopted for the transitive propagation of change in collaborative business processes [5, 9].

As a result, the mechanism described by Alg. 1 returns a set of new monitoring capabilities that must be created to guarantee the alignment between monitoring and compliance requirements. Note that some of such monitoring capabilities may have already been created to preserve alignment of previously introduced compliance requirements. The information returned by the alignment mechanism is used by the designer to:

```
/* main procedure                                                    */
Get blocks on which new_cr predicates;
foreach block do
    run single alignment check of new_cr, block;
end

/* single alignment check of new_cr, block                           */
if block is not outsourced then
    Calculate monitoring requirements induced by new_cr on block;
    Create monitoring capabilities to satisfy identified monitoring
    requirements;
else
    find block in outsourced process;
    /* recursive call                                                */
    run single alignment check of new_cr, block;
end
```

Algorithm 1: Alignment with monitoring requirements of a new compliance requirement new_cr

- request the creation of missing monitoring capabilities to the actors who can provide them;
- connect existing monitoring capabilities to new compliance requirements, to guarantee their verification at runtime.

For lack of space, we omit the algorithmic representation of the mechanism to ensure alignment when a compliance requirement is deleted. In principle, a compliance requirement may be deleted without the need for any additional actions. However, we also argue that each actor in a BN should disclose only the minimal amount of information required to monitor compliance. Therefore, when a compliance requirement is deleted, all its required monitoring capabilities may also be deleted. Deleting monitoring capabilities is suggested by our framework only if they are not required by other compliance requirements in the BN.

Alg. 2 presents the mechanism to align monitoring and compliance requirement when a block of a process is outsourced. The mechanism simply runs the *single alignment check* defined by Alg. 1 on all compliance requirements predicating on the block that has been outsourced.

For lack of space, we omit the algorithmic representation of the mechanism to ensure alignment in the case of backsourcing. Similarly to what described for outsourcing, the monitorability of each existing compliance requirememt involving a block that is backsourced should be rechecked as new.

```
/* outsourcing alignment mechanism                                    */
Get list of compliance requirements cr predicating on block;
foreach cr do
 | run single alignment check of cr, block;
end
```

Algorithm 2: Alignment of monitoring and compliance requirements in case of outsourcing of a block

4 Concretization for Specific Implementation Choices

As introduced before, in order to capture a real world scenario, a concretization step is required (see Fig.1). Concretization involves the specification of the following aspects:

- the format of compliance and monitoring requirements;
- the framework chosen for modelling business processes (and, specifically, outsourcing and backsourcing).

Once the above aspects are specified, it would be possible for a designer to use our conceptual model and alignment mechanisms to model a BN and manage the alignment of compliance and monitoring requirements as the BN evolves.

In this section we discuss a set of choices for the *concretization* of our conceptual framework. Specifically, the implementation choices described here are the ones implemented by the prototype described in Section 5. This discussion is also provided as a reference example to support designers who may need to accommodate different implementation choices for the concretization of our conceptual model and alignment mechanisms.

We first discuss the derivation of monitoring requirements and capabilities for a specific way of expressing compliance requirements and our choice to model outsourcing based on an extension of BPMN in Section 4.1. Then, in Section 4.2, we derive a concrete implementation of the alignment algorithms presented in Section 2 based on our concretization choices. For illustration purposes, we also discuss an example of an evolving BN in the healthcare industry in Section 4.3.

4.1 Concretization of Monitoring Capabilities and Outsourcing Model

Compliance requirements can be specified using a variety of languages, such as event calculus [21], deontic logic [23], or LTL [8]. Similarly, business process outsourcing in structured processes can be modelled using different notations, such as BPMN, BPEL, or structured EPCs [13].

We consider the set of atomic compliance patterns adopted in [8], which are reported in the first column of Table 1. Patterns involve one or two blocks as operands. Note that arbitrarily complex (composite) compliance requirements may be derived by combining the atomic patterns of Table 1 using standard

Compliance Pattern	Description	Occr(·)	Tmst(·)	Role(·)	OtEv(·)
exists A	A must occur in the BN	A	×	×	×
A precedes B	B must always be preceded by A	A, B	A, B	×	×
A leadsTo B	A must always be followed by B	A, B	A, B	×	×
A segrFrom B	A and B assigned to different roles	×	×	A, B	×
A inclusive B	Presence of A mandates presence of B	A, B	×	×	×
A prerequisite B	Absence of A mandates absence of B	×	×	×	A, B
A exclusive B	Presence of A mandates absence of B (and vice versa)	A, B	×	×	A, B
A substitute B	Presence of A substitute absence of B	A	×	×	B
A corequisite B	Either A and B are preesent together, or absent together	A, B	×	×	A, B

Table 1. Atomic Compliance Requirements Patterns and Required Monitoring Capabilities.

logical operators. As pattern composition does not introduces any specificity to our approach, we will not consider it further in this paper.

Table 1 classifies the monitoring capabilities required by the compliance requirements considered in this work and matches these with the compliant requirements patterns. Given the monitoring requirements, from a technical standpoint a monitoring capability becomes a point of access, i.e. an interface, for interested stakeholders, such as auditors or compliance experts, to access the information implied by the monitoring requirement. For example, in order to check the satisfaction of the compliance requirement exists A, a proof of the occurrence of A is required, for instance in the form of an information system log showing the execution of A in at least one process instance. This is information has to be made available by the actor executing A, for instance through an operation of a monitoring Web service.

As far as outsourcing is concerned, we extend our conceptual model as depicted in Fig. 3(a). We consider three types of blocks, namely internal, outsourced, and placeholder blocks [5]. Internal blocks are executed internally (*in-house*) by one specific actor. The actor may be able to provide the required monitoring capabilities involving such a block. Placeholders blocks can only be part of outsourced blocks and are introduced to maintain a link between the process structure and the related compliance requirements. Fig. 3 presents also a generic example of business process outsourcing. In Fig. 3(b) the actor actor1 executes all required tasks in house, i.e. all blocks of the business process are internal. Fig. 3(c) shows the resulting BN when blocks B and C are outsourced to actor2. After outsourcing, the process executed by actor1 includes an outsourced block BC_OUT (represented, with an abuse of notation, as a BPMN expanded process in the figure), which will reference the process executed by actor2. The outsourced block includes placeholder blocks bearing the same name as the internal blocks that have been outsourced, namely B and C. These are required to keep the consistency between compliance requirements and the blocks they

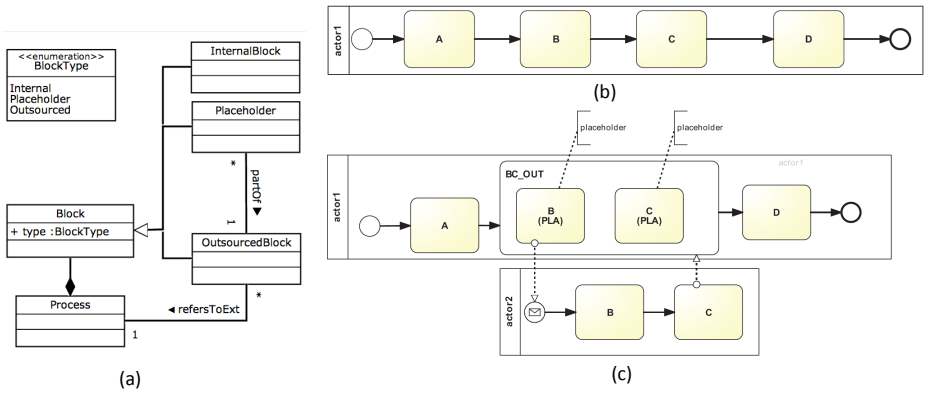

Fig. 3. Outsourcing modelling and generic example

predicate on. If, in fact, blocks B and C would be simply removed after outsourcing, then it would be impossible to know that compliance requirements originally predicating on such blocks should now refer to the related internal blocks of the process of actor2.

Note that, for the sake of simplicity, we also make the hypothesis that the internal blocks of the process executed by the outsourcer, i.e. actor2, bear the same name as the internal blocks originally outsourced (and now become placeholders). Generally, string equality may be substituted by any mapping function between the labels chosen for placeholders and the correspondent internal blocks in the referenced outsourced process.

4.2 Concretization of Alignment Mechanisms

In this section we describe the concrete mechanisms implementing the conceptual alignment mechanims described by Alg. 1 and Alg. 2. The concretization is based on the implementation choices regarding compliance requirements, monitoring capabilities, and outsourcing discussed in the previous section.

Alg. 3 shows the mechanism required to preserve alignment when a new compliance requirement cr is added to the BN. The mechanism returns a set of actions required to ensure alignment, i.e. a set of monitoring capabilities monCapSET that should be implemented. The main procedure PreserveMonitorability() calls a sub-procedure checkMonitorability() on all operands of cr (see line 5). The sub-procedure computes the required monitoring capabilities for each internal block on which cr predicates (line 3). If a required monitoring capability is not currently available in the BN, then its creation is added to the list of actions required to ensure monitorability (lines 4-8). If an operand is a placeholder, then the sub-procedure is recursively called on the correspondent internal block of the process to which the placeholder is outsourced (line 13). Such a recursive call allows the functioning of the mechanisms on arbitrary long outsourcing *chains* in the BN.

```
1: procedure:: PreserveMonitorability(cr:ComplianceRequirement)
2: monCapSET:MonitoringCapability[] ← ∅ {create empty list of monitoring
   capabilities}
3: opSET:Block[] ← cr.predicatesOn
4: for all blk ∈ opSET do
5:    monCapSET ← monCapSET ∪ checkMonitorability(cr, blk, monCapSET)
6: end for
7: return monCapSET
8: end procedure
```

```
1: procedure:: checkMonitorability(cr:ComplianceRequirement, blk:Block,
   monCapSET:MonitoringCapability[])
2: if blk.type = INT then
3:    mrSET:MonitoringRequirement[] ← cr.induce{Get induced monitoring
      requirements}
4:    for all mr ∈ mrSET do
5:       if ∄ a: MonitoringCapability : mr.satisfiedBy=a ∧ a.refersTo=blk then
6:          monCapSET ← mc:MonitoringCapability: mr.satisfiedBy = mc ∧
            mc.refersTo=blk {Add required monitorig capability to list}
7:       end if
8:    end for
9: end if
10: if blk.type = PLA then
11:    pro:Process ← blk.partOf.refersToExt {get process to which block is
       outsourced}
12:    iblk:InternalBlock ← b: b.name = blk.name ∧ b ∈ pro {find block in
       outsourced process}
13:    checkMonitorability(cr, iblk, MonCapSet) {recursive call}
14: end if
15: return monCapSET
```

Algorithm 3: Concrete alignment for new compliance requirements cr

The concrete mechanism ensuring alignment in case of outsourcing is shown in Alg. 4. It takes as input a set of blocks BLK to be outsourced and a reference extPro to the external process where these have to be outsourced. The mechanism first checks whether outsourcing is feasible, that is, whether the external process has blocks bearing the same name as the internal blocks to be outsourced (lines 4-6). Note that in this work we are not concerned with the internal semantic equivalence of tasks, but we only consider mapping equivalence among task labels to operate the internal/placeholder block substitution.

Then, the mechanism adjusts the structural aspects of the BN. In particular, the internal block(s) to be outsourced should be encapsulated into an outsourced block and each of them replaced by a placeholder bearing the same name (lines 7-12). A reference from the created outsourced block to the external process

```
1: procedure:: Outsource(BLK:Block[],extPro:Process)
2: isFeasible:boolean ← true
3: for all blk ∈ BLK do
4:   if ∄ b:Block s.t. b ∈ extPro ∧ b.name = blk.name then
5:     isFeasible ← false; break;
6:   end if
7:   if isFeasible then
8:     TMP: Block[] ← BLK {Save blocks to outsource for later...}
9:     newOutBlk:OutsourcedBlock ← new OutsourcingBlock()
10:    newOutBlk.refersToExt ← extPro {Set ref. to outsourcing process}
11:    for all blk ∈ BLK do
12:      blk ← new Placeholder(); {Create placeholder in place of internal
         block}
13:      blk.partOf ← newOutBlk {set ref. to oustourced block}
14:    end for
15:    monCapSET:MonitoringCapability[] ← ∅
16:    for all cr:ComplianceRequirement s.t. cr.predicatesOn ⊆ TMP do
17:      monCapSET ← monCapSET ∪ PreserveMonitorability(cr) {Restore
         monitorability of involved compliance requirements}
18:    end for
19:  else
20:    !! Outsourcing not feasible !!
21:  end if
22: end for
```

Algorithm 4: Concrete alignment in case of outsourcing

should also be set (line 13). Once the structure of business processes in the BN has been updated correctly, a monitorability check should run for all compliance requirements involving the blocks outsourced (lines 15-17). As a result of outsourcing, in fact, the blocks involved by some compliance requirements may have changed from internal to placeholders. This may bear an impact on their monitorability. Therefore, each compliance requirement is now treated as new (hence, the procedure PreserveMonitorability() described in Algorithm 3 is called).

4.3 Example

As an example, we consider an adaptation of the healthcare business network considered in [16], depicted in Fig. 4. Initially, the network involves two actors, i.e. Hospital and Lab, and two processes, i.e. Patient Treatment and Blood Test. The two actors are not collaborating at the moment, as there is no relationship between the two processes. Fig. 4 also shows two compliance requirements for the process contributed by the Hospital. The first requirement relates to segregation of duties, that is, for quality reasons the doctor performing the discharge should differ from the doctor who examined the patient. The second requirement predicates that, for privacy reasons, blood samples should always be destroyed

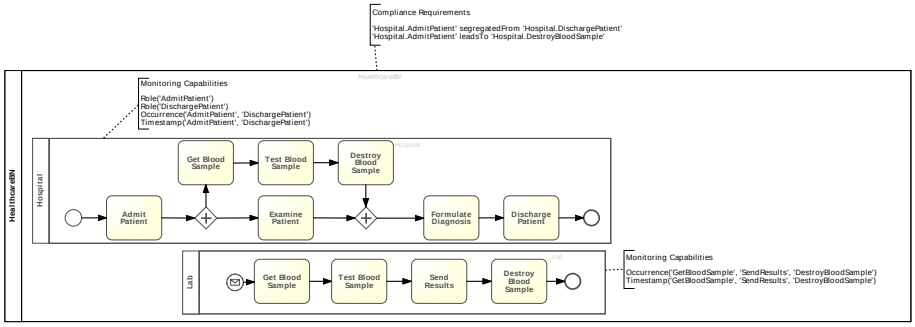

Fig. 4. Running Example: Healthcare BN

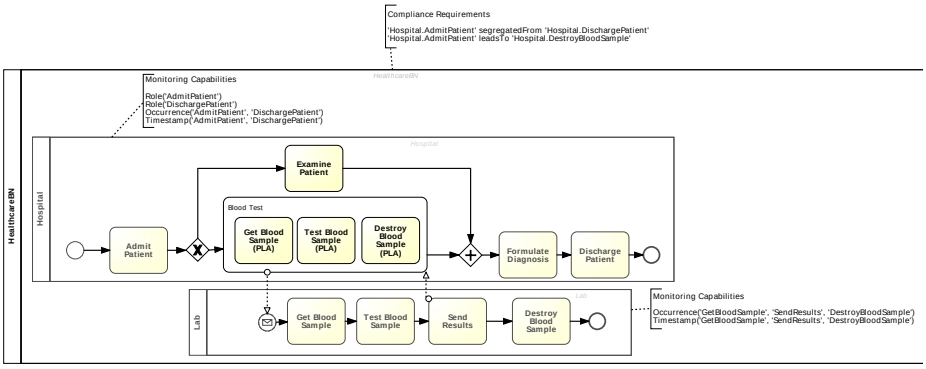

Fig. 5. Outsourcing blood testing in the running example

after being tested. Note that all blocks in the processes of Fig. 4 (represented as BPMN tasks) are internal.

Given the monitoring capabilities of the Hospital reported in Fig. 4, only the first compliance requirement can currently be monitored. Monitoring the second requirement requires occurrence and timestamp capabilities over the block DestroyBloodSample, which are currently not available for the Hospital.

Fig. 5 shows the BN resulting from the outsourcing by the Hospital of the blood testing to the Lab. Because of outsourcing, the framework recognizes that the task DestroyBloodSample for the Hospital becomes a placehodler for a task that has been outsourced to the Lab. The outsourcing changes monitoring requirements of the BN and, therefore, their alignment with the compliance requirements. In this specific case, the compliance requirements are now aligned, as the monitoring capabilities required to monitor the compliance requirements that could not be provided by the Hospital before outsourcing (see Fig. 4), can now be provided to interested stakeholders by the Lab. In other words, the application of the mechanism of Algorithm 4 returns an empty set of actions, as all

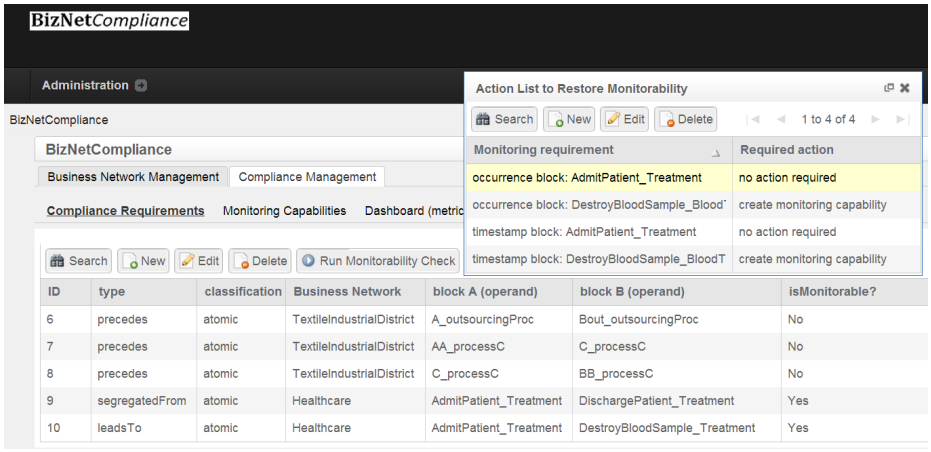

Fig. 6. List of actions to restore compliance monitorability in BizNetCompliance.

required monitoring capabilities to monitor the compliance requirements defined in the BN are now available.

5 Prototype Implementation

Our framework for alignment of process compliance and monitoring requirements is implemented in the prototype BizNetCompliance [1]. It allows the business process expert to specify the structure of BNs, in terms of actors, business processes, and related blocks. The compliance expert can specify the compliance and monitoring requirements in the BN and run the alignment mechanisms in case of BN evolution.

The core of the application is constituted by two components to manage business networks structure and compliance requirements. The tool can be extended by plug-ins. A plug-in we developed allows the evaluation of compliance alignment metrics, which we briefly describe later in this section.

As far as network structure is concerned, the tool allows business process experts to specify the aspects of business processes relevant for compliance, i.e. the block structure, excluding control flow and connectors; alternatively, the tool can extract the block structure from processes specified in BPMN. The extensions to BPMN required by our framework, i.e. to specify outsourced blocks and placeholders, exploit the extension mechanism of the *Task* element proposed in BPMN 2.0.

BizNetCompliance is a Web-based information system that can be accessed using a browser. It has been partly realized using a state of the art model-driven tool for enterprise systems application development and integration[2].

[1] `https://sites.google.com/site/biznetcompliancemonitor/`

[2] `www.mendix.com`

Fig. 6 shows a snapshot of the output of the tool after the execution of the mechanism in Alg. 3 to preserve alignment when the compliance requirement AdmitPatient leadsTo DestroyBloodSample is introduced in the BN of our running example. The output in this case is constituted by a set of monitoring capabilities currently missing in the BN, which should be implemented to preserve the alignment between the new compliance requirement and its related monitoring requirements (see Fig. 4).

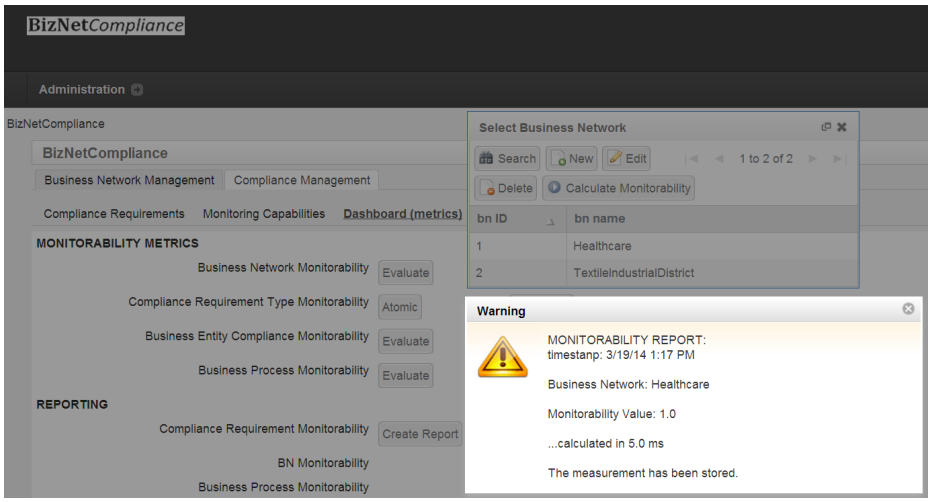

Fig. 7. Dashboard for compliance monitorability and example report in BizNetCompliance.

As anticipated, we extended the core of the tool with a plug-in to evaluate compliance-monitoring alignment metrics. Such metrics give a quantitative evaluation of the degree of alignment between compliance and monitoring requirements and are important for a variety of reasons. In the design phase, and in respect of network evolution, alignment metrics are important *ex-ante*, i.e. to assess the impact on the BN of a particular evolution, and *ex-post*, i.e. to assess the status of the BN from the point of view monitorability after evolution has occurred. In the ex-ante perspective, alignment metrics can help evaluating, for instance, the impact of outsourcing a given process. While outsourcing may be beneficial from an economic point of view, it may also disrupt the compliance monitorability of the BN in case blocks involved by many compliance requirements are outsourced to a business entity with limited monitoring capabilities. In the ex-post perspective, alignment metrics give a synthetic view of the compliance level of a BN. Higher values of such metrics are likely to increase the appeal of the BN to potential customers or potential business partners.

The basic metric we consider is the monitorability of an individual compliance requirement $MON(cr)$. Given the number D of monitoring capabilities induced

(desired) by a compliance requirement and the number A of such capabilities currently available within the BN where the compliance reuqirements is required, we have $MON(cr) = A/D$, with $0 \leq MON(cr) \leq 1$, $\forall\, cr$.

Given the basic metric, we defined metrics at the level of business network, processes, entities, and type of compliance requirement:

- Business network alignment: average monitorability of all compliance requirements defined in a BN;
- Business process alignment: average monitorability of all compliance requirements involving a particular process, i.e. using at least one of the blocks of the process as operand;
- Actor compliance alignment: average monitorability of all compliance requirements involving all processes contributed by a given business entity (actor);
- Compliance requirement type alignment: average monitorability of all compliance requirements of a given type.

The metrics do not aim at being exhaustive, but they rather aim at giving a glimpse of the potential of our framework in giving a quantitative perspective about the compliance-monitoring alignment in a BN.

Figure 7 shows a snapshot of the dashboard to monitor the compliance monitorability metrics of a BN and an example evaluation. The system also enables users to print detailed reports of the monitorability gaps in the BN, e.g. list of missing monitoring capabilities to achieve full monitorability of existing compliance requirements.

We evaluated the performance of our tool in respect of the time required to (i) execute the alignment algorithms and (ii) calculate the values of metrics defined above. The execution of the outsourcing alignment mechanism of Alg. 4 is the most critical from a performance point of view, as its execution time grows polynomially with the number of compliance requirements predicating on the outsourced block(s) and the depth level of the outsourcing chain. We created a testbed involving example processes with up to 20 outsourced activities and up to 5-level deep outsourcing chains, with complex compliance requirements obtained as combination of the atomic patterns of Table 1 predicating on up to 20 blocks (activities). On a standard desktop machine equipped with a quad-core CPU our tool returns the list of monitoring capabilities to be created to maintain alignment in less than 50ms. Note that the runtime performance of our tool is not particularly critical as the tool is intended to be used at design time.

6 Related Work

Compliance management of internal business processes has received a large deal of attention recently in the areas of compliance requirements specification [23], monitoring [10], and diagnostics [8, 22]. A review of compliance management frameworks can be found in [7]. A review of emerging research challenges in

compliance management is presented in [1]. Among others, the authors identify (i) difficulties in creating evidence of compliance, (ii) frequent changes in regulations, and (iii) lack of IT support as major challenges for companies implementing compliance. We address the above challenges in the context of cross-organizational compliance management by (i) providing a framework to assess the monitoring capabilities required to guarantee compliance when (ii) new compliance requirements are deployed. We show the feasibility of our approach by (iii) discussing a prototype implementation.

Concerning compliance management in collaborative scenarios, Reichert et al. [16] identify the challenges of compliance management in cross-organizational processes. A subset of such challenges concerns the dynamic aspect of BNs and, in particular, the need for a compliance management framework to adapt to changes of processes and requirements within BNs. This paper contributes towards closing this gap, by considering the alignment between compliance and monitoring requirements for specific types of change, i.e. out/backsourcing and deployment of new compliance requirements.

Our literature review identified three main contributions specific to the case of compliance management in collaborative scenarios [15, 20, 24].

The issue of cross-organizational compliance is tackled using interaction models in [15]. The authors propose a method to guarantee global compliance requirements on a collaboration by looking at the interaction models derived by matching the process public views of collaborating parties. Compliance is satisfied if the requirements fit the generated interaction models. The approach deals with static collaborations, as it does not explicitly consider either requirements or collaboration structural changes.

The work presented in [20] defines a model for the event-based compliance checking of contract-based collaborations. Although the work focuses on business conversations, i.e. the messages exchanged during a collaboration, rather than business processes, it proposes concepts that we have adopted in our framework. Similarly to the monitoring capabilities of our framework, each business operation used in a conversation should generate a set of monitoring events to allow the compliance checker to run compliance control. Moreover, events are timestamped, to check those properties referring to the order of messages and operations.

A declarative way to model cross-organizational processes is based on the notion of *commitments* of the agents involved in the collaboration [24, 6]. Commitments may be used to specify and verify compliance requirements [24]. This approach prevents compliance to predicate on the structural aspects of business networks and fits the case in which formalized business process models are not available to describe the collaboration.

As far as compliance metrics are concerned, the paper [17] defines the metric of design-time compliance distance, where the structure of a business process is assessed against the ideal situation represented by the satisfaction of compliance requirements. In our framework, compliance is ideal when all the monitoring ca-

pabilities required by a compliance requirement are available within the business network.

7 Conclusions and Outlook

In this paper we presented a framework for aligning compliance and monitoring requirements in evolving business networks. The framework comprises a conceptual model of business networks, compliance and monitoring requirements, and their evolution and mechanisms to ensure alignment of compliance and monitoring requirements. We discussed a *concretization* of the framework for specific implementation choices regarding the modelling of compliance requirements and business process outsourcing. Such implementation choices have driven the prototype implementation.

A first extension of this work will concern the evaluation of our framework with practitioners in a real world BN. In particular, we aim at evaluating (i) the extent to which our framework is able to cover all relevant compliance and monitoring requirements in the chosen scenario and (ii) the reduced effort for the compliance expert in managing the complexity of evolving requirements. Such an evaluation will involve working sessions and possibly controlled experiments with practitioners actively engaging with our prototype implementation in modelling compliance requirements and assessing the impact of evolution on the monitorability of a BN. Specifically, we will consider the case of a BN of mobility services that we have already engaged for testing tools for formulating and communicating network-based service strategy [18].

The models proposed in this paper can be extended, by looking at new types of network or compliance evolution, and by considering the run-time view of our framework, e.g. monitoring of executing processes, sampling of relevant compliance data, root-cause analysis of compliance requirements violations. The alignmemt metrics allow the definition of enhanced BN management features, such as dynamic partner selection or process delegation based on the maximization of the alignment between process compliance and monitoring requirements.

References

1. N. S. Abdullah, S. Sadiq, and M. Indulska. Emerging challenges in information systems research for regulatory compliance management. In *Proc. CAiSE*, pages 251–265, 2010.
2. D. Ardagna, L. Baresi, S. Comai, M. Comuzzi, , and B. Pernici. A service-based framework for flexible business processes. *IEEE Software*, 28(2):61–67, 2011.
3. S.-M.-R. Behesti, B. Benatallah, and H. Motahari-Nezhad. Enabling the analysis of cross-cutting aspects in ad-hoc processes. In *Proc. CAiSE*, pages 51–67, 2013.
4. L. Camarinha-Matos and H. Afsarmanesh. A comprehensive modeling framework for collaborative networked organizations. *Journal of Intelligent Manufacturing*, 18(5):529–542, 2007.
5. M. Comuzzi, J. Vonk, and P. Grefen. Measures and mechanisms for process monitoring in evolving business networks. *Data Knowl. Eng.*, 71:1–28, 2012.

6. N. Desi, A. K. Chopra, and M. P. Singh. Amoeba: A methodology for modeling and evolving cross-organizational business processes. *ACM TOSEM*, 29(2), 2009.
7. M. El Kharbili. Business process regulatory compliance management solution frameworks: A comparative evaluation. In *APCCM*, pages 23–32, 2012.
8. A. Elgammal, O. Turetken, W.-J. van den Heuvel, and M. Papazoglou. Root-cause analysis of design-time compliance viiolations on the basis of property patterns. In *Proc. ICSOC*, pages 17–31, 2010.
9. W. Fdhila, S. Rinderle-Ma, and M. Reichert. Change propagation in collaborative processes scenarios. In *Proc. IEEE CollaborateCom*, pages 452–461, 2012.
10. C. Giblin, S. Müller, , and B. Pfitzmann. From regulatory policies to event monitoring rules: Towards model-driven compliance automation. Technical Report 99682, IBM Research GmbH, Z'urich, 2006.
11. P. Grefen, R. Eshuis, N. Mehandijev, G. Kouvas, and G. Weichart. Internet-based support for process-oriented instant virtual enterpises. *IEEE Internet Comput.*, pages 30–38, 2009.
12. P. Grefen, H. Ludwig, and S. Angelov. A three-level framework for process and data management of complex E-Services. *IJCIS*, 12(4):487–531, 2003.
13. P. Grefen, H. Ludwig, A. Dan, and S. Angelov. An analysis of web services support for dynamic business process outsourcing. *Information & Software Technology*, 48(11):1115–1134, 2006.
14. D. Karagiannis, J. Mylopoulos, and M. Schwab. Business process-based regulation compliance: The case of the sarbanes-oxley act. In *IEEE Int. Requirements Engineering Conference*, pages 315–321, 2007.
15. D. Knuplesch, M. Reichert, W. Fdhila, , and S. Rinderle-Ma. On enabling compliance of cross-organizational business processes. In *Proc. BPM*, pages 146–154, 2013.
16. D. Knuplesch, M. Reichert, J. Mangler, S. Rinderle-Ma, and W. Fdhila. Towards compliance of cross-organizational processes and their changes. In *BPM Workshops*, pages 649–661, 2013.
17. R. Lu, S. Sadiq, and G. Governatori. Measurement of compliance distance in business processes. *Information Systems Management*, 25:344–355, 2008.
18. E. Lüftenegger, M. Comuzzi, and P. Grefen. The service-dominant ecosystem: Mapping a service dominant strategy to a product-service ecosystem. In *Proc. PRO-VE*, pages 22–30, 2013.
19. J. Mendling, H. Reijers, and W. van der Aalst. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2), 2010.
20. C. Molina-Jimenez, S. Shrivastava, and M. Strano. A model for checking contractual compliance of business interactions. *IEEE Transactions on Services Computing*, 5(2):276–289, 2012.
21. M. Montali, F. Maggi, F. Chesani, P. Mello, and W. van der Aalst. Monitoring business constraints with the event calculus. In *ACM Transactions on Intelligent Systems and Technology*, volume 5, 2013.
22. E. Ramezani, D. Fahland, and W. van der Aalst. Where did i misbehave? diagnostic information in compliance checking. In *Proc. BPM*, pages 262–278, 2012.
23. S. Sadiq, G. Governatori, and K. Namiri. Modeling control objectives for business process compliance. In *Proc. BPM*, pages 149–164, 2007.
24. P. Telang and M. Singh. Specifying and verifying cross-organizational business models: An agent-oriented approach. *IEEE Transactions on Services Computing*, 5(3):305 – 318, 2012.